Yee & Associates, P.C.

4100 Alpha Road
Suite 1100
Dallas, Texas 75244

Main No. (972) 385-8777
Facsimile (972) 385-7766

Facsimile Cover Sheet

To: Commissioner for Patents for **Examiner Lechi Truong** **Group Art Unit 2194**	Facsimile No.: **571/273-8300**
From: Carrie Parker Legal Assistant to Vicky Ash	No. of Pages Including Cover Sheet: **9**

Message:

Enclosed herewith:

- Transmittal of Reply Brief; and
- Reply Brief.

Re: Application No. **09/583,411**
Attorney Docket No: **AUS000153US1**

Date: Monday, April 17, 2006

Please contact us at (972) 385-8777 if you do not receive all pages indicated above or experience any difficulty in receiving this facsimile.

PLEASE CONFIRM RECEIPT OF THIS TRANSMISSION BY FAXING A CONFIRMATION TO 972-385-7766.

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application of: **Taylor**

Serial No.: **09/583,411**

Filed: **May 31, 2000**

For: **Method and Apparatus for Bridging Service of Standard Object Identifier Based Protocols**

§ Group Art Unit: **2194**
§
§ Examiner: **Truong, Lechi**
§
§ Attorney Docket No.: **AUS000153US1**
§

35525
PATENT TRADEMARK OFFICE
CUSTOMER NUMBER

Certificate of Transmission Under 37 C.F.R. § 1.8(a)
I hereby certify this correspondence is being transmitted via facsimile to the Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450, facsimile number (571) 273-8300 on April 17, 2006.

By: Carrie Parker
Carrie Parker

TRANSMITTAL OF REPLY BRIEF

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:
ENCLOSED HEREWITH:

- Reply Brief (37 C.F.R. 41.41).

No fees are believed to be required. If, however, any fees are required, I authorize the Commissioner to charge these fees which may be required to IBM Corporation Deposit Account No. 09-0447.

Respectfully submitted,

Gerald H. Glanzman
Registration No. 25,035
Duke W. Yee
Registration No. 34,285
YEE & ASSOCIATES, P.C.
P.O. Box 802333
Dallas, Texas 75380
(972) 385-8777
ATTORNEYS FOR APPLICANTS

RECEIVED
CENTRAL FAX CENTER

APR 17 2006

Docket No. AUS000153US1 *PATENT*

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application of: **Taylor**

Serial No. **09/583,411**

Filed: **May 31, 2000**

For: **Method and Apparatus for Bridging Service of Standard Object Identifier Based Protocols**

Group Art Unit: **2194**

Examiner: **Truong, Lechi**

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

35525
PATENT TRADEMARK OFFICE
CUSTOMER NUMBER

Certificate of Transmission Under 37 C.F.R. § 1.8(a)
I hereby certify this correspondence is being transmitted via facsimile to the Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450, facsimile number (571) 273-8300 on April 17, 2006.

By: Carrie Parker
Carrie Parker

REPLY BRIEF (37 C.F.R. 41.41)

This Reply Brief is submitted in response to the Examiner's Answer mailed on February 16, 2006.

No fees are believed to be required to file a Reply Brief. If any fees are required for filing this Reply Brief, those fees are dealt with in the accompanying TRANSMITTAL OF REPLY BRIEF.

SUMMARY OF CLAIMED SUBJECT MATTER

Appellants thank Examiner Lechi Truong for the courtesies extended to Appellants' representatives during the April 4, 2006 telephone call. The Examiner confirmed that the summary of claimed subject matter is correct.

RESPONSE TO EXAMINER'S REMARKS

A. GROUND OF REJECTION 1 (Claims 1, 9, 20, and 39)

Claims 1, 9, 20 and 39 are rejected under 35 U.S.C. § 103(a) as being allegedly unpatentable over *Spofford et al.* (U.S. Patent 5,913,037), hereinafter referred to as *Spofford*, in view of *Dobbins et al.* (U.S. Patent 5,951,649), hereinafter referred to as *Dobbins* and further in view of *Pearson* (U.S. Patent 6,023,684). This rejection is respectfully traversed.

Claim 1, which is representative of the other rejected independent claims 20 and 39 with regard to similarly recited subject matter, is reproduced below for the convenience of the Board:

> 1. A method on a server in a distributed data processing system for maintaining a logical composite repository of Object Identifier (OID) tree structures, the method comprising the steps of:
>
> receiving, in an OID abstraction layer, an OID tree structure from a repository; wherein the OID abstraction layer is capable of receiving queries for objects in two or more different protocols and supports the two or more different protocols by mapping queries from multiple protocol interfaces to application program interface (API) requests that the repository understands;
>
> registering the OID tree structure with a registry associated with the OID abstraction layer; and
>
> adding the OID tree structure to a repository associated with the OID abstraction layer.

Claim 9 is also reproduced below for the convenience of the Board:

> 9. A method on a server in a distributed data processing system for retrieving object data from a repository, comprising:
>
> receiving a first query for the object data from a requester in the distributed data processing system, wherein the first query is in a protocol recognized by an OID abstraction layer; wherein the OID abstraction layer is capable of receiving queries for objects in two or more different protocols and supports the two or more different protocols by mapping queries from multiple protocol interfaces to application program interface (API) requests that the repository understands;
>
> interpreting the first query according to the protocol recognized by the OID abstraction layer, wherein the protocol recognized by the OID abstraction layer is one of the two or more different protocols;
>
> locating a repository that contains the object data requested in the first query based on a registry associated with the OID abstraction layer; and
>
> retrieving the object data from the repository using an OID abstraction layer application program interface (API).

Appellants continue to submit that *Spofford*, *Dobbins*, and *Pearson*, either taken individually or in combination, do not teach or suggest that "the OID abstraction layer is capable of receiving queries for objects in two or more different protocols and supports the two or more different protocols by mapping queries from multiple protocol interfaces to application program interface (API) requests that the repository understands," as recited in independent claims 1, 20, and 39. Further, Appellants continue to submit that *Spofford*, *Dobbins*, and *Pearson*, either taken individually or in combination, do not teach or suggest "receiving a first query for the object data from a requester in the distributed data processing system, wherein the first query is in a protocol recognized by an OID abstraction layer; wherein the OID abstraction layer is capable of receiving queries for objects in two or more different protocols and supports the two or more different protocols by mapping queries from multiple protocol interfaces to application program interface (API) requests that the repository understands," as recited in independent claim 9.

In the Examiner's Answer, the Examiner states the following in responding to Appellants' arguments:

> **As to the point (1) and (2)** Dobbin teaches the OID abstraction layer in capable of receiving queries for objects in two or more different protocols, the first query is in protocol recognized by OUD abstraction layer (a standard interface for the Management Information Base (MIB) for object access by any management protocol or other entity including SNMP, SNMPv2, DMP, col 16, In 20-23, The MIB object provide the mapping of object Identifier, col 17, In 19-23 I the specific Managed Objects: name: OID Managed object, col 17, In 51-64/ col 19, In 30-35). The Form of these requests are composed of queries to an object within database, by using the object's identifier (OID).
>
> Pearson teaches supports the two or more different protocols by mapping queries from multiple protocol interfaces to application program interface (API) requests that the repository understands (convert data from a parsed client request to a format compatible with the API for the application service identified in the application service call, col 11, In 15-20/ converting client messages between the language supported by the client program and the language used to implement a application service, col 4, In 67 to col 5, In 1-3/ converts user queries to the appropriate query language format for the, col 2, In 60-65/ presentation logic 80 communication with client program using HTML documents, other communication protocols may be used, col 11, In 42-45/ client messages which are in the format of a known internet service, such as E-mail, Files transfer protocol, col 5, In 60-65/ col 10, In 32-37).
>
> **As to the point (3),** Dobbin teaches Simple network Management Protocol (SNMP) operates by passing request to MIB. The form of these request are composed of queries to object within database, by using the object (OID), col 29, In 33-38), the IBM Navigator simplifier the format of these requests by proving a textual representation to these OID's (col 29, In 40-45).

Examiner's Answer dated February 16, 2006, page 11-12.

As discussed previously in the Appeal Brief, *Dobbins* discloses that SNMP or any other management protocols may be used to access an object using a standard interface for the MIB, but does not teach that multiple management protocols may be used to access an object. *Dobbins* only shows an SNMP agent in Figures 3A and 3B, and discloses using SNMP in the preferred embodiment of *Dobbins*. *Dobbins* discloses that a MIB Navigator simplifies the request from SNMP by providing a textual representation (ASCII names) to an OID, which is easier for the user to digest than a string of numerals. *Dobbins* does not teach or suggest that "the OID abstraction layer is capable of receiving queries for objects in two or more different protocols and supports the two or more different protocols by mapping queries from multiple protocol interfaces to application program interface (API) requests that the repository understands."

The Examiner refers to portions of the following section of *Dobbins* in the rejection of the independent claims:

> Specific Managed Objects 66
>
> The MIB Object 65 provides the mapping of an Object Identifier (which uniquely identifies a piece of data in the Management Information Base) with a particular Managed Object. It then takes advantage of the Managed Object Base Class standard interface to retrieve the actual data held in the Managed Object.
>
> The Core Objects 61 represent a set of objects used by the entire MOF. These include one object for each of the nine basic types defined in the SMI: INTEGER 67, OCTET STRING 68, NULL 69, OBJECT IDENTIFIER 70, IPAddress 71, Counter 72, Gauge 73, TimeTicks 74, and Opaque 75. Each of these objects encompass the functionality and valid states described in the SMI, as well as additional functionality useful to the code which uses these types. Other Core Objects include Table Objects 64 for holding the data for Managed Object tables.
>
> The Managed Object Base Class 62 provides a standard method of access to specific Managed Objects 66. Any object which derives from this base class need only customize a small amount of code to allow this standard access, which the MIB then takes advantage of in obtaining the object's data.
>
> The Specific Managed Objects 66 are data types that are multiply derived from the Core Objects 61 and the Managed Object Base Class 62 to encompass the functionality of each. For instance, an Integer Managed Object contains the integer functionality derived from the INTEGER Core Object 67, as well as the manageability derived from the Managed Object Base Class 62. When a new Integer Managed Object is needed, a designer simply needs to provide the value of the integer (or information on how to obtain the value) and a unique Object Identifier. This requires relatively little new code to accomplish. The MIB Object 65 is then able to access the Managed Object in a standard way. Managed Object Tables are also created in this way. The following is an

enumeration of the specific Managed Objects:

Name	Derived From
Integer Managed Object	Core INTEGER, Managed Object Base Class
String Managed Object	Core OCTET STRING, Managed Object Base Class
OID Managed Object	Core OBJECT IDENTIFIER, Managed Object Base Class
Null Managed Object	Core NULL, Managed Object Base Class

Dobbins, column 17, lines 19-64.

This section of *Dobbins* teaches that MIB Object 65 provides the mapping of an Object Identifier (which uniquely identifies a piece of data in the Management Information Base) with a particular Managed Object. This section also teaches that when a new Integer Managed Object is needed, a designer simply needs to provide the value of the integer (or information on how to obtain the value) and a unique Object Identifier and that this requires relatively little new code to accomplish. The MIB Object 65 is then able to access the Managed Object in a standard way. *Dobbins* does not teach or suggest "the OID abstraction layer is capable of receiving queries for objects in two or more different protocols and supports the two or more different protocols by mapping queries from multiple protocol interfaces to application program interface (API) requests that the repository understands."

In view of the above and the Appeal Brief filed on November 14, 2005, Appellants respectfully submit that claims 1-57 are allowable over the cited prior art and that the application is in condition for allowance. Accordingly, Appellants respectfully request the Board of Patent Appeals and Interferences to not sustain the rejections set forth in the Final Office Action and in the Examiner's Answer.

Gerald H. Glanzman
Reg. No. 25,035
YEE & ASSOCIATES, P.C.
PO Box 802333
Dallas, TX 75380
(972) 385-8777

GHG/vja